EXHIBIT W

TO DECLARATION OF S. MERRILL WEISS IN SUPPORT OF PLAINTIFF ACACIA MEDIA TECHNOLOGIES CORPORATION'S MEMORANDUM OF POINTS AND AUTHORITIES IN OPPOSITION TO ROUND 3 DEFENDANTS' MOTION FOR SUMMARY JUDGMENT OF INVALIDITY UNDER 35 U.S.C. § 112 OF THE '992, '863, AND '702 PATENTS; AND SATELLITE DEFENDANTS' MOTION FOR SUMMARY JUDGMENT OF INVALIDITY OF THE '992, '863, AND '720 PATENTS

Hovsepian v. Apple, Inc. Doc. 325 Att. 23

Dockets.Justia.com

An Algorithm for Vector Quantizer Design

YOSEPH LINDE, MEMBER, IEEE, ANDRÉS BUZO, MEMBER, IEEE, AND ROBERT M. GRAY, SENIOR MEMBER, IEEE

***Abstract*—An efficient and intuitive algorithm is presented for the design of vector quantizers based either on a known probabilistic model or on a long training sequence of data. The basic properties of the algorithm are discussed and demonstrated by examples. Quite general distortion measures and long blocklengths are allowed, as exemplified by the design of parameter vector quantizers of ten-dimensional vectors arising in Linear Predictive Coded (LPC) speech compression with a complicated distortion measure arising in LPC analysis that does not depend only on the error vector.**

INTRODUCTION

An efficient and intuitive algorithm for the design of good block or vector quantizers with quite general distortion measures is developed for use on either known probabilistic source descriptions or on a long training sequence of data. The algorithm is based on an approach of Lloyd [1], is not a variational technique, and involves no differentiation; hence it works well even when the distribution has discrete components, as is the case when a sample distribution obtained from a training sequence is used. As with the common variational techniques, the algorithm produces a quantizer meeting necessary but not sufficient conditions for optimality. Usually, however, at least local optimality is assured in both approaches.

We here motivate and describe the algorithm and relate it to a number of similar algorithms for special cases that have appeared in both the quantization and cluster analysis literature. The basic operation of the algorithm is simple and intuitive in the general case considered here and it is clear that variational techniques are not required to develop nor to apply the algorithm.

Several of the algorithm's basic properties are developed using heuristic arguments and demonstrated by example. In a companion theoretical paper [2], these properties are given precise mathematical statements and are proved using arguments from optimization theory and ergodic theory. Those results will occasionally be quoted here to characterize the generality of certain properties.

Paper approved by the Editor for Data Communication Systems of the IEEE Communications Society for publication after presentation in part at the 1979 International Telemetering Conference, Los Angeles, CA, November 1978 and the 1979 International Symposium on Information Theory, Gringano, Italy, June 1979. Manuscript received May 22, 1978; revised August 21, 1979. This work was supported by Air Force Contract F44620-73-0065, F49620-78-C-0087, and F49620-79-C-0058 and by the Joint Services Electronics Program at Stanford University, Stanford, CA.

Y. Linde is with the Codex Corporation, Mansfield, MA.

A. Buzo was with Stanford University, Stanford, CA and Signal Technology Inc., Santa Barbara, CA. He is now with the Instituto de Ingenieria, National University of Mexico, Mexico City, Mexico.

R. M. Gray is with the Information Systems Laboratory, Stanford University, Stanford, CA 94305.

In particular, the algorithm's convergence properties are demonstrated herein by several examples. We consider the usual test case for such algorithms—quantizer design for memoryless Gaussian sources with a mean-squared error distortion measure; but we design and evaluate block quantizers with a rate of one bit per symbol and with blocklengths of 1 through 6. Comparison with recently developed lower bounds to the optimal distortion of such block quantizers (which provide strict improvement over the traditional bounds of rate-distortion theory) indicate that the resulting quantizers are indeed nearly optimal and not simply locally optimal. We also consider a scalar case where local optima arise and show how a variation of the algorithm yields a global optimum.

The algorithm is also used to design a quantizer for 10-dimensional vectors arising in speech compression systems. A complicated distortion measure is used that does not simply depend on the error vector. No probabilistic model is assumed, and hence the quantizer must be designed based on a training sequence of real speech. Here the convergence properties for both length of the training sequence and the number of iterations of the algorithm are demonstrated experimentally. No theoretical optimum is known for this case, but our system was used to compress the output of a traditional 6000 bit/s Linear Predictive Coded (LPC) speech system down to a rate of 1400 bits/s with only a slight loss in quality as judged by untrained listeners in informal subjective tests. To the authors' knowledge, direct application of variational techniques have not succeeded in designing block quantizers for such large block lengths and such complicated distortion measures.

BLOCK QUANTIZERS

An N-level k-dimensional quantizer is a mapping, q, that assigns to each input vector, $x = (x_0, \cdots, x_{k-1})$, a reproduction vector, $\hat{x} = q(x)$, drawn from a finite reproduction alphabet, $\hat{A} = \{y_i; i = 1, \cdots, N\}$. The quantizer q is completely described by the reproduction alphabet (or codebook) $\hat{A}$ together with the partition, $S = \{S_i; i = 1, \cdots, N\}$, of the input vector space into the sets $S_i = \{x: q(x) = y_i\}$ of input vectors mapping into the i^{th} reproduction vector (or codeword). Such quantizers are also called block quantizers, vector quantizers, and block source codes.

DISTORTION MEASURES

We assume the distortion caused by reproducing an input vector x by a reproduction vector $\hat{x}$ is given by a nonnegative distortion measure $d(x, \hat{x})$. Many such distortion measures have been proposed in the literature. The most common for

0090-6778/80/0100-0084$00.75 © 1980 IEEE

Exhibit W Page 434

reasons of mathematical convenience is the squared-error distortion.

$$d(x,\hat{x}) = \sum_{i=0}^{k-1} |x_i - \hat{x}_i|^2 \tag{1}$$

Other common distortion measures are the l_ν, or Holder norm,

$$d(x,\hat{x}) = \left\{ \sum_{i=0}^{k-1} |x_i - \hat{x}_i|^\nu \right\}^{1/\nu} \triangleq \|x - \hat{x}\|_\nu, \tag{2}$$

and its ν^{th} power, the ν^{th}-law distortion:

$$d(x,\hat{x}) = \sum_{i=0}^{k-1} |x_i - \hat{x}_i|^\nu = \|x - \hat{x}\|_\nu^\nu. \tag{3}$$

While both distortion measures (2) and (3) depend on the ν^{th} power of the errors in the separate coordinates, the measure of (2) is often more useful since it is a distance or metric and hence satisfies the triangle inequality, $d(x, \hat{x}) \leqslant d(x, y) + d(y, \hat{x})$, for all y. The triangle inequality allows one to bound the overall distortion easily in a multi-step system by the sum of the individual distortions incurred in each step. The usual ν^{th}-law distortion of (3) does not have this property. Other distortion measures are the l_∞, or Minkowski norm,

$$d(x,\hat{x}) = \max_{0 \leqslant i \leqslant k-1} |x_i - \hat{x}_i|, \tag{4}$$

the weighted-squares distortion,

$$d(x,\hat{x}) = \sum_{i=0}^{k-1} w_i |x_i - \hat{x}_i|^2, \tag{5}$$

where $w_i \geqslant 0$, $i = 0, \cdots, k-1$, and the more general quadratic distortion

$$d(x,\hat{x}) = (x - \hat{x})B(x - \hat{x})^t = \sum_{i=0}^{k-1} \sum_{j=0}^{k-1} B_{i,j}(x_i - \hat{x}_i)(x_j - \hat{x}_j), \tag{6}$$

where $B = \{B_{i,j}\}$ is a $k \times k$ positive definite symmetric matrix.

All of the previously described distortion measures have the property that they depend on the vectors x and $\hat{x}$ only through the error vector $x - \hat{x}$. Such distortion measures having the form $d(x, \hat{x}) = L(x - \hat{x})$ are called difference distortion measures. Distortion measures not having this form but depending on x and $\hat{x}$ in a more complicated fashion have also been proposed for data compression systems. Of interest here is a distortion measure of Itakura and Saito [3, 4] and Chaffee [5, 32] which arises in speech compression systems and has the form

$$d(x,\hat{x}) = (x - \hat{x})R(x)(x - \hat{x})^t, \tag{7}$$

where for each x, $R(x)$ is a positive definite $k \times k$ symmetric matrix. This distortion resembles the quadratic distortion of (6), but here the weighting matrix depends on the input vector x.

We are here concerned with the particular form and application of this distortion measure rather than its origins, which are treated in depth in [3-9] and in a paper in preparation. For motivation, however, we briefly describe the context in which this distortion measure is used in speech systems. In the LPC approach to speech compression [10], each frame of sampled speech is modeled as the output of a finite-order all-pole filter driven by either white noise (unvoiced sounds) or a periodic pulse train (voiced sounds). LPC analysis has, as input, a frame of speech and produces parameters describing the model. These parameters are then quantized and transmitted. One collection of such parameters consists of a voiced/unvoiced decision together with a pitch estimate for voiced sounds, a gain term σ (related to volume), and the sample response of the normalized inverse filter $(1, a_1, a_2, \cdots, a_K)$, that is, the normalized all-pole model has transfer function or z-transform $\{\sum_{k=0}^{K} a_k z^{-k}\}^{-1}$. Other parameter descriptions such as the reflection coefficients are also possible [10].

In traditional LPC systems, the various parameters are quantized separately, but such systems have effectively reached their theoretical performance limits [11]. Hence it is natural to consider block quantization of these parameters and compare the performance with the traditional scalar quantization techniques. Here we consider the case where the pitch and gain are (as usual) quantized separately, but the parameters describing the normalized model are to be quantized together as a vector. Since the lead term is 1, we wish to quantize a vector $(a_1, a_2, \cdots, a_K) \triangleq x = (x_0, \cdots, x_{K-1})$. A distortion measure, $d(x, \hat{x})$, between x and a reproduction x, can then be viewed as a distortion measure between two normalized (unit gain) inverse filters or models. A distortion measure for such a case has been proposed by Itakura and Saito [3, 4] and by Chaffee [5, 32] and it has the form of (7) with $R(x)$ the autocorrelation matrix $\{r_x(k-j); k = 0, 1, \cdots, K-1; j = 0, 1, \cdots, K-1\}$ defined by

$$r_x(k) = \int_{-\pi}^{\pi} \left| \sum_{m=0}^{K} a_i e^{-im\theta} \right|^{-2} e^{ik\theta} d\theta / 2\pi, \tag{8}$$

described by x when the input has a flat unit amplitude spectrum.

Many properties and alternative forms for this particular distortion measure are developed in [3-9], where it is also shown that standard LPC systems implicitly minimize this distortion, which suggests that it is also an appropriate distortion measure for subsequent quantization. Here, however, the important fact is that it is not a difference distortion measure; it is one for which the dependence on x and $\hat{x}$ is quite complicated.

We also observe that various functions of the previously defined distortion measures have been proposed in the literature, for example, distortion measures of the forms $\|x - \hat{x}\|^r$ and $\rho(\|x - \hat{x}\|)$, where ρ is a convex function and the norm is any of the previously defined norms. The techniques to be developed here are applicable to all of these distortion measures.

Exhibit W Page 435

PERFORMANCE

Let $X = (X_0, \cdots, X_{k-1})$ be a real random vector described by a cumulative distribution function $F(x) = \Pr\{X_i \leq x_i; i = 0, 1, \cdots, k-1\}$. A measure of the performance of a quantizer q applied to the random vector X is given by the expected distortion

$$D(q) = Ed(X, q(X)), \tag{9}$$

where E denotes the expectation with respect to the underlying distribution F. This performance measure is physically meaningful if the quantizer q is to be used to quantize a sequence of vectors $X_n = (X_{nK}, \cdots, X_{nK+K-1})$ that are stationary and ergodic, since then the time-averaged distortion,

$$n^{-1} \sum_{i=0}^{n-1} d(X_i, q(X_i)),$$

converges with probability one to $D(q)$ as $n \to \infty$ (from the ergodic theorem), that is, $D(q)$ describes the long-run time-averaged distortion.

An alternative performance measure is the maximum of $d(x, q(x))$ over all x in A, but we use only the expected distortion (9) since, in most problems of interest (to us), it is the average distortion and not the peak distortion that determines subjective quality. In addition, the expected distortion is more easily dealt with mathematically.

OPTIMAL QUANTIZATION

An N-level quantizer will be said to be optimal (or globally optimal) if it minimizes the expected distortion, that is, q^* is optimal if for all other quantizers q having N reproduction vectors $D(q^*) \leq D(q)$. A quantizer is said to be locally optimum if $D(q)$ is only a local minimum, that is, slight changes in q cause an increase in distortion. The goal of block quantizer design is to obtain an optimal quantizer if possible and, if not, to obtain a locally optimal and hopefully "good" quantizer. Several such algorithms have been proposed in the literature for the computer-aided design of locally optimal quantizers. In a few special cases, it has been possible to demonstrate global optimality either analytically or by exhausting all local optima. In 1957, in a classic but unfortunately unpublished Bell Laboratories' paper, S. Lloyd [1] proposed two methods for quantizer design for the scalar case ($k = 1$) with a squared-error distortion criterion. His "Method II" was a straightforward variational approach wherein he took derivatives with respect to the reproduction symbols, y_i, and with respect to the boundary points defining the S_i and set these derivatives to zero. This in general yields only a "stationary-point" quantizer (a multidimensional zero derivative) that satisfies necessary but not sufficient conditions for optimality. By second derivative arguments, however, it is easy to establish that such stationary-point quantizers are at least locally optimum for ν^{th}-power law distortion measures. In addition, Lloyd also demonstrated global optimality for certain distributions by a technique of exhaustively searching all local optima. Essentially the same technique was also proposed and used in the parallel problem of cluster analysis by Dalenius [12] in 1950, Fisher [13] in 1953, and Cox [14] in 1957. The technique was also independently developed by Max [15] in 1960 and the resulting quantizer is commonly known as the Lloyd-Max quantizer. This approach has proved quite useful for designing scalar quantizers, with power-law distortion criteria and with known distributions that were sufficiently well behaved to ensure the existence of the derivatives in question. In addition, for this case, Fleischer [16] was able to demonstrate analytically that the resulting quantizers were globally optimum for several interesting probability densities.

In some situations, however, the direct variational approach has not proved successful. First, if k is not equal to 1 or 2, the computational requirements become too complex. Simple combinations of one-dimensional differentiation will not work because of the possibly complicated surface shapes of the boundaries of the cells of the partition. In fact, the only successful applications of a direct variational approach to multidimensional quantization are for quantizers where the partition cells are required to have a particular simple form such as multidimensional "cubes" or, in two dimensions, "pie slices," each described only by a radius and two angles. These shapes are amenable to differentiation techniques, but only yield a local optimum within the constrained class. Secondly, if, in addition, more complex distortion measures such as those of (4)-(7) are desired, the required computation associated with the variational equations can become exorbitant. Thirdly, if the underlying probability distribution has discrete components, then the required derivatives may not exist, causing further computational problems. Lastly, if one lacks a precise probabilistic description of the random vector X and must base the design instead on an observed long training sequence of data, then there is no obvious way to apply the variational approach. If the underlying unknown process is stationary and ergodic, then hopefully a system designed by using a sufficiently long training sequence should also work well on future data. To directly apply the variational technique in this case, one would first have to estimate the underlying continuous distribution based on the observations and then take the appropriate derivatives. Unfortunately, however, most statistical techniques for density estimation require an underlying assumption on the class of allowed densities, e.g., exponential families. Thus these techniques are inappropriate when no such knowledge is available. Furthermore, a good fit of a continuous model to a finite-sample histogram may have ill-behaved differential behavior and hence may not produce a good quantizer. To our knowledge, no one has successfully used such an approach nor has anyone demonstrated that this approach will yield the correct quantizer in the limit of a long training sequence.

Lloyd [1] also proposed an alternative nonvariational approach as his "Method I." Not surprisingly, both approaches yield the same quantizer for the special cases he considered, but we shall argue that a natural and intuitive extension of his Method I provides an efficient algorithm for the design of good vector quantizers that overcomes the problems of the variational approach. In fact, variations of Lloyd's Method I have been "discovered" several times in the literature for

Exhibit W Page 436

squared-error and magnitude-error distortion criteria for both scalar and multidimensional cases (e.g., [22], [23], [24], [31]). Lloyd's basic development, however, remains the simplest, yet it extends easily to the general case considered here.

To describe Lloyd's Method I in the general case, we first assume that the distribution is known, but we allow it to be either continuous or discrete and make no assumptions requiring the existence of derivatives. Given a quantizer q described by a reproduction alphabet $\hat{A} = \{y_i; i = 1, \cdots, N\}$ and partition $S = \{S_i; i = 1, \cdots, N\}$, then the expected distortion $D(\{\hat{A}, S\}) \triangleq D(q)$ can be written as

$$D(\{\hat{A}, S\}) = Ed(X, q(X))$$

$$= \sum_{i=1}^{N} E(d(X, y_i) \mid X \in S_i) \Pr(X \in S_i), \qquad (10)$$

where $E(d(X, y_i) \mid X \in S_i)$ is the conditional expected distortion, given $X \in S_i$, or, equivalently, given $q(X) = y_i$.

Suppose that we are given a particular reproduction alphabet $\hat{A}$, but a partition is not specified. A partition that is optimum for $\hat{A}$ is easily constructed by mapping each x into the $y_i \in \hat{A}$ minimizing the distortion $d(x, y_i)$, that is, by choosing the minimum distortion or nearest-neighbor codeword for each input. A tie-breaking rule such as choosing the reproduction with the lowest index is required if more than one codeword minimizes the distortion. The partition, say $P(\hat{A}) = \{P_i; i = 1, \cdots, N\}$ constructed in this manner is such that $x \in P_i$ (or $q(x) = y_i$) only if $d(x, y_i) \leqslant d(x, y_j)$, all j, and hence

$$D(\{\hat{A}, P(\hat{A})\}) = E(\min_{y \in \hat{A}} d(X, y)) \qquad (11)$$

which, in turn, implies for any partition S that

$$D(\{\hat{A}, S\}) \geqslant D(\{\hat{A}, P(\hat{A})\}). \qquad (12)$$

Thus for a fixed reproduction alphabet $\hat{A}$, the best possible partition is $P(\hat{A})$.

Conversely, assume we are given a partition $S = \{S_i; i = 1, \cdots, N\}$ describing a quantizer. For the moment, assume also that the distortion measure and distribution are such that, for each set S with nonzero probability in k-dimensional Euclidean space, there exists a minimum distortion vector $\hat{x}(S)$ for which

$$E(d(X, \hat{x}(S)) \mid X \in S) = \min_{u} E(d(X, u) \mid X \in S). \qquad (13)$$

Analogous to the case of a squared-error distortion measure and a uniform probability distribution, we call the vector $\hat{x}(S)$ the centroid or center of gravity of the set S. If such points exist, then clearly for a fixed partition $S = \{S_i; i = 1, \cdots, N\}$, no reproduction alphabet $\hat{A} = \{y_i; i = 1, \cdots, N\}$ can yield a smaller average distortion then the reproduction alphabet $\hat{x}(S) \triangleq \{\hat{x}(S_i); i = 1, \cdots, N\}$ containing the centroids of the sets in S since

$$D(\{\hat{A}, S\}) = \sum_{i=1}^{N} E(d(X, y_i) \mid X \in S_i) \Pr(X \in S_i)$$

$$\geqslant \sum_{i=1}^{N} \min_{u} E(d(X, u) \mid X \in S_i) \Pr(X \in S_i)$$

$$= D(\{\hat{x}(S), S\}). \qquad (14)$$

It is shown in [2] that the centroids of (13) exist for all sets S with nonzero probability for quite general distortion measures including all of those considered here. In particular, if $d(x, y)$ is convex in y, then centroids can be computed using standard convex programming techniques as described, e.g., in Luenberger [17, 18] or Rockafellar [19]. In certain cases, they can be found easily using variational techniques. If the probability of a set S is zero, then the centroid can be defined in an arbitrary manner since then the conditional expectation given that S in (13) has no unique definition.

Equations (12) and (14) suggest a natural algorithm for designing a good quantizer by taking any given quantizer and iteratively improving it:

Algorithm (Known Distribution)

(0) Initialization: Given N = number of levels, a distortion threshold $\epsilon \geqslant 0$, and an initial N-level reproduction alphabet $\hat{A}_0$ and a distribution F. Set $m = 0$ and $D_{-1} = \infty$.

(1) Given $\hat{A}_m = \{y_i; i = 1, \cdots, N\}$, find its minimum distortion partition $P(\hat{A}_m) = \{S_i; i = 1, \cdots, N\}$: $x \in S_i$ if $d(x, y_i) \leqslant d(x, y_j)$ for all j. Compute the resulting average distortion, $D_m = D(\{\hat{A}_m, P(\hat{A}_m)\}) = E \min_{y \in \hat{A}_m} d(X, y)$.

(2) If $(D_{m-1} - D_m)/D_m \leqslant \epsilon$, halt with $\hat{A}_m$ and $P(\hat{A}_m)$ describing final quantizer. Otherwise continue.

(3) Find the optimal reproduction alphabet $\hat{x}(P(\hat{A}_m)) = \{\hat{x}(S_i); i = 1, \cdots, N\}$ for $P(\hat{A}_m)$. Set $\hat{A}_{m+1} \triangleq \hat{x}(P(\hat{A}_m))$. Replace m by $m + 1$ and go to (1).

If, at some point, the optimal partition $P(\hat{A}_m)$ has a cell S_i such that $\Pr(X \in S_i) = 0$, then the algorithm, as stated, assigns an arbitrary vector as centroid and continues. Clearly, alternative rules are possible and may perform better in practice. For example, one can simply remove the cell S_i and the corresponding reproduction symbol from the quantizer without affecting performance, and then continue with an $(N - 1)$ level quantizer. Alternatively, one could assign to S_i its Euclidean center of gravity or the i^{th} centroid from the previous iteration. One could also simply reassign the reproduction vector corresponding to S_i to another cell S_j and continue the algorithm. The stated technique is given simply for convenience, since zero probability cells were not a problem in the examples considered here. They can, however, occur and in such situations alternative techniques such as those described may well work better. In practice, a simple alternative is that, if the final quantizer produced by the algorithm has a zero probability (hence useless) cell, simply rerun the algorithm with a different initial guess.

Exhibit W Page 437

From (12) and (14), $D_m \leqslant D_{m-1}$ and hence each iteration of the algorithm must either reduce the distortion or leave it unchanged. We shall later mention some minor additional details of the algorithm and discuss techniques for choosing an initial guess, but the previously given description contains the essential ideas.

Since D_m is nonincreasing and nonnegative, it must have a limit, say D_∞, as $m \to \infty$. It is shown in [2] that if a limiting quantizer $\hat{A}_\infty$ exists in the sense $\hat{A}_m \to \hat{A}_\infty$ as $m \to \infty$ in the usual Euclidean sense, then $D(\{\hat{A}_\infty, P(\hat{A}_\infty)\}) = D_\infty$ and $\hat{A}_\infty$ has the property that $\hat{A}_\infty = \hat{x}(P(\hat{A}_\infty))$, that is, $\hat{A}_\infty$ is exactly the centroid of its own optimal partition. In the language of optimization theory, $\{\hat{A}_\infty, P(\hat{A}_\infty)\}$ is a *fixed point* under further iterations of the algorithm [17, 18]. Hence the limit quantizer (if it exists) is called a fixed-point quantizer (in contrast to a stationary-point quantizer obtained by a variational approach). In this light, the algorithm is simply a standard technique for finding a fixed point via the method of successive approximation (see, e.g., Luenberger [17, p. 272]). If $\epsilon = 0$ and the algorithm halts for finite m, then such a fixed point has been attained [2].

It is shown in [2] that a necessary condition for a quantizer to be optimal is that it be a fixed-point quantizer. It is also shown in [2] that, as in Lloyd's case, if a fixed-point quantizer is such that there is no probability on the boundary of the partition cells, that is if $\Pr(d(X, y_i) = d(X, y_j)$, for some $i \neq j) = 0$, then the quantizer is locally optimum. This is always the case with continuous distributions, but can in principle be violated for discrete distributions. It was never found to occur in our experiments, however. As Lloyd suggests, the algorithm can easily be modified to test a fixed point for this condition and if there is nonzero probability of a vector on a boundary, the strategy would be to reassign the vector to another cell of the partition and continue the iteration.

For the $N = 1$ case with a squared-error distortion criterion, the algorithm is simply Lloyd's Method I, and his arguments apply immediately in the more general case considered herein. A similar technique was earlier proposed in 1953 by Fisher [13] in a cluster analysis problem using Bayes decisions with a squared-error cost. For larger dimensions and distortion measures of the form $d(x, \hat{x}) = \|x - \hat{x}\|_2^r$, $r \geqslant 1$, the relations (12) and (14) were observed by Zador [20] and Gersho [21] in their work on the asymptotic performance of optimal quantizers, and hence the algorithm is certainly implicit in their work. They did not, however, actually propose or apply the technique to design a quantizer for fixed N. In 1965, Forgy [31] proposed the algorithm for cluster analysis for the multidimensional squared-error distortion case and a sample distribution (see the discussion in MacQueen [25]). In 1977, Chen [22] proposed essentially the same algorithm for the multidimensional case with the squared-error distortion measure and used it to design two-dimensional quantizers for vectors uniformly distributed in a circle.

Since the algorithm has no differentiability requirements, it is valid for purely discrete distributions. This has an important application to the case where one does not possess *a priori* a probabilistic description of the source to be compressed, and hence must base his design on an observed long training sequence of the data to be compressed. One approach would be to use standard density estimation techniques of statistics to obtain a "smooth" distribution of F and to then apply variational techniques. As previously discussed, we do not adopt this approach as it requires additional assumptions on the allowed densities. Instead we consider the following approach: Use the training sequence, say $\{x_k; k = 0, \cdots, n-1\}$ to form the time-average distortion

$$\frac{1}{n} \sum_{i=0}^{n-1} d(x_i, q(x_i))$$

and observe that this is exactly the expected distortion $E_{G_n} d(X, q(X))$ with respect to the sample distribution G_n determined by the training sequence, i.e., the distribution that assigns probability m/n to a vector x that occurs in the training sequence m times. Thus we can design a quantizer that minimizes the time-average distortion for the training sequence by running the algorithm on the sample distribution G_n (1,2). This yields the following variation of the algorithm:

Algorithm (Unknown Distribution)

(0) Initialization: Given N = number of levels, distortion threshold $\epsilon \geqslant 0$, an initial N-level reproduction alphabet $\hat{A}_0$, and a training sequence $\{x_j; j = 0, \cdots, n-1\}$. Set $m = 0$ and $D_{-1} = \infty$.

(1) Given $\hat{A}_m = \{y_i; i = 1, \cdots, N\}$, find the minimum distortion partition $P(\hat{A}_m) = \{S_i; i = 1, \cdots, N\}$ of the training sequence: $x_j \in S_i$ if $d(x_j, y_i) \leqslant d(x_j, y_l)$, for all l. Compute the average distortion

$$D_m = D(\{\hat{A}_m, P(\hat{A}_m)\}) = n^{-1} \sum_{j=0}^{n-1} \min_{y \in \hat{A}_m} d(x_j, y).$$

(2) If $(D_{m-1} - D_m)/D_m \leqslant \epsilon$, halt with $\hat{A}_m$ final reproduction alphabet. Otherwise continue.

(3) Find the optimal reproduction alphabet $\hat{x}(P(\hat{A}_m)) = \{\hat{x}(S_i); i = 1, \cdots, N\}$ for $P(\hat{A}_m)$. Set $\hat{A}_{m+1} \triangleq \hat{x}(P(\hat{A}_m))$. Replace m by $m + 1$ and go to (1).

Observe above that while designing the quantizer, only partitions of the training sequence (the input alphabet) are considered. Once the final codebook $\hat{A}_m$ is obtained, however, it is used on new data outside the training sequence with the optimum nearest-neighbor rule, that is, an optimum partition of k-dimensional Euclidean space.

(1) It was observed by a reviewer that application of the algorithm to the sample distribution provides a "Monte Carlo" design of the quantizer for a vector with a known distribution, that is, the design is based on samples of the random vectors rather than on an explicit distribution.

(2) During the period this paper was being reviewed for publication, two similar techniques were reported for special cases. In 1978, Capria, Westin, and Esposito [23] presented a similar technique for the scalar case using dynamic programming arguments. Their approach was for average squared-error distortion and for maximum distortion over the training sequence. In 1979, Menez, Boeri, and Esteban [24] proposed a similar technique for scalar quantization using squared-error and magnitude-error distortion measures.

Exhibit W Page 438

If the sequence of random vectors is stationary and ergodic, then it follows from the ergodic theorem that, with probability one, G_n goes to the true underlying distribution F as $n \to \infty$. Thus if the training sequence is sufficiently long, hopefully a good quantizer for the sample distribution G_n should also be good for the true distribution F, and hence should yield good performance for future data produced by the source. All of these ideas are made precise in [2] where it is shown that, subject to suitable mathematical assumptions, the quantizer produced by applying the algorithm to G_n converges, as $n \to \infty$, to the quantizer produced by applying the algorithm to the true underlying distribution F. We observe that no analogous results are known to the authors for the density estimation/variational approach, and that independence of successive blocks is not required for these results, only block stationarity and ergodicity.

We also point out that, for finite alphabet distributions such as sample distributions, the algorithm always converges to a fixed-point quantizer in a finite number of steps [2].

A similar technique used in cluster analysis with squared-error cost functions was developed by MacQueen in 1967 [25] and has been called the k-means approach. A more involved technique using the k-means approach is the "ISODATA" approach of Ball and Hall [26]. The basic idea of finding minimum distortion partitions and centroids is the same, but the training sequence data is used in a different manner and the resulting quantizers will, in general, be different. Their sequential technique incorporates the training vectors one at a time and ends when the last vector is incorporated. This is in contrast to the previous algorithm which considers all of the training vectors at each iteration. The k-means method can be described as follows: The goal is to produce a partition $S_0 = \{S_0, \cdots, S_{N-1}\}$ of the training alphabet, $A = \{x_i; i = 0, \cdots, n-1\}$ consisting of all vectors in the training sequence. The corresponding reproduction alphabet $\hat{A}$ will then be the collection of the Euclidean centroids of the sets S_i, that is, the final reproduction alphabet will be optimal for the final partition (but the final partition may not be optimal for the final reproduction alphabet, except as $n \to \infty$). To obtain S, we first think of the each S_i as a bin in which to place training sequence vectors until all are placed. Initially, we start by placing the first N vectors in separate bins, i.e., $x_i \in S_i$, $i = 0, \cdots, N-1$. We then proceed as follows: at each iteration, a new training vector x_m is observed. We find the set S_i for which the distortion between x_m and the centroid $\hat{x}(S_i)$ is minimized and then add x_m to this bin. Thus, at each iteration, the new vector is added to the bin with the closest centroid, and hence the next time, this bin will have a new centroid. This operation is continued until all sample vectors are incorporated.

Although similar in philosophy, the k-means algorithm has some crucial differences. In particular, it is suited for the case where *only* the training sequence is to be classified, that is, where a long sequence of vectors is to be grouped in a low distortion manner. The sequential procedure is computationally efficient for grouping, but a "quantizer" is not produced until the procedure is stopped. In other words, in cluster analysis, one wishes to group things and the groups can change with time, but in quantization, one wishes to fix the groups (to get a time-invariant quantizer), and then use these groups (or the quantizer) on future data outside of the training sequence.

An additional problem is that the only theorems which guarantee convergence, in the limit of a long training sequence, require the assumption that successive vectors be independent [25], unlike the more general case for the proposed algorithm [2].

Recently Levenson *et al.* used a variation of the k-means and ISODATA algorithms with a distortion measure proposed by Itakura [4] to determine reference templates for speaker-independent word recognition [27]. They used, as a distortion measure, the logarithm of the distortion of (7) (which is a gain-optimized Itakura-Saito distortion [7]—our use of the distortion measure with unit-gain-normalized models results in no such logarithmic function). In their technique, however, a minimax rule was used to select the reproduction vectors (or cluster points) rather than finding the "optimum" centroid vector. If instead, the distortion measure of (7) is used, then the centroids are easily found, as will be seen.

CHOICE OF $\hat{A}_0$

There are several ways to choose the initial reproduction alphabet $\hat{A}_0$ required by the algorithm. One method for use on sample distributions is that of the k-means method, namely choosing the first N vectors in the training sequence. We did not try this approach as, intuitively, one would like these vectors to be well-separated, and N consecutive samples may not be. Two other methods were found to be useful in our examples. The first is to use a uniform quantizer over all or most of the source alphabet (if it is bounded). For example, if used on a sample distribution, one uses a k-dimensional uniform quantizer on a k-dimensional Euclidean cube including all or most of the points in the training sequence. This technique was used in the Gaussian examples described later.

The second technique is useful when one wishes to design quantizers of successively higher rates until achieving an acceptable level of distortion. Here we consider M-level quantizers with $M = 2^R$, $R = 0, 1, \cdots$, and continue until we achieve an initial guess for an N-level quantizer as follows:

INITIAL GUESS BY "SPLITTING"

(0) Initialization: Set $M = 1$ and define $\hat{A}_0(1) = \hat{x}(A)$, the centroid of the entire alphabet (the centroid of the training sequence, if a sample distribution is used).

(1) Given the reproduction alphabet $\hat{A}_0(M)$ containing M vectors $\{y_i; i = 1, \cdots, M\}$, "split" each vector y_i into two close vectors $y_i + \epsilon$ and $y_i - \epsilon$, where ϵ is a fixed perturbation vector. The collection $\tilde{A}$ of $\{y_i + \epsilon, y_i - \epsilon, i = 1, \cdots, M\}$ has $2M$ vectors. Replace M by $2M$.

(2) Is $M = N$? If so, set $\hat{A}_0 = \tilde{A}(M)$ and halt. $\tilde{A}_0$ is then the initial reproduction alphabet for the N-level quantization algorithm. If not, run the algorithm for an M-level quantizer on $\tilde{A}(M)$ to produce a good reproduction alphabet $\hat{A}_0(M)$, and then return to step (1).

Using the splitting algorithm on a training sequence, one starts with a one-level quantizer consisting of the centroid of the training sequence. This vector is then split into two vectors

Exhibit W Page 439

and the two-level quantizer algorithm is run on this pair to obtain a good (fixed-point) two-level quantizer. Each of these two vectors is then split and the algorithm is run to produce a good four-level quantizer. At the conclusion, one has fixed-point quantizers for $1, 2, 4, 8, \cdots, N$ levels.

EXAMPLES

Gaussian Sources

The algorithm was used initially to design quantizers for the classical example of memoryless Gaussian random variables with the squared-error distortion criterion of (1), based on a training sequence of data. The training and sample data were produced by a zero-mean, unit-variance memoryless sequence of Gaussian random variables. The initial guess was a unit quantizer on the k-dimensional cube, $\{x: |x_i| \leq 4; i = 0, \cdots, k-1\}$. A distortion threshold of 0.1% was used. The overall algorithm can be described as follows:

(0) Initialization: Fix N = number of levels, k = block length, n = length of training sequence, $\epsilon = .001$. Given a training sequence $\{x_j; j = 0, \cdots, n-1\}$. Let $\hat{A}_0$ be an N-level uniform quantizer reproduction alphabet for the k-dimensional cube, $\{u: |u_i| \leq 4, i = 0, \cdots, k-1\}$. Set $m = 0$ and $D_{-1} = \infty$.

(1) Given $\hat{A}_m = \{y_i; i = 1, \cdots, N\}$, find the minimum-distortion partition $P(\hat{A}_m) = \{S_i; i = 1, \cdots, N\}$. For example, for each $j = 0, \cdots, n-1$, compute $d(x_j, y_i)$ for $i = 1, \cdots, N$. If $d(x_j, y_i) \leq d(x_j, y_l)$ for all l, then $x_j \in S_i$. Compute:

$$D_m = D(\{\hat{A}_m, P(\hat{A}_m)\}) = n^{-1} \sum_{j=0}^{n-1} \min_{y \in \hat{A}_m} d(x_j, y).$$

(2) If $(D_{m-1} - D_m)/D_m \leq \epsilon = .001$, halt with final quantizer described by $\hat{A}_m$. Otherwise continue.

(3) Find the optimal reproduction alphabet $\hat{x}(P(\hat{A}_m)) = \{\hat{x}(S_i); i = 1, \cdots, N\}$ for $P(\hat{A}_m)$. For the squared-error criterion, $\hat{x}(S_i)$ is the Euclidean center of gravity or centroid given by

$$x(S_i) = \frac{1}{\|S_i\|} \sum_{j: x_j \in S_i} x_j,$$

where $\|S_i\|$ denotes the number of training vectors in the cell S_i. If $\|S_i\| = 0$, set $\hat{x}(S_i) = y_i$, the old codeword. Define $\hat{A}_{m+1} = \hat{x}(P(A_m))$, replace m by $m + 1$, and go to (1).

Table 1 presents a simple but nontrivial example intended to demonstrate the basic operation of the algorithm. A two-dimensional quantizer with four levels is designed, based on a short training sequence of twelve training vectors. Because of the short training sequence in this case, the final distortion is lower than one would expect and the final quantizer may not work well on new data outside of the training sequence. The tradeoffs between the length of the training sequence and the performance inside and outside the training sequence are developed more carefully in the speech example.

Observe that, in the example of Table 1, the algorithm could actually halt in step (1) of the $m = 1$ iteration since, if $P(\hat{A}_m) = P(\hat{A}_{m-1})$, it follows that $\hat{A}_{m+1} = \hat{x}(P(\hat{A}_m)) = \hat{x}(P(\hat{A}_{m-1})) = \hat{A}_m$, and hence $\hat{A}_m$ is the desired fixed point.

TABLE 1
A SIMPLE EXAMPLE

(0) Initialization: $N = 4$, $k = 2$, $\epsilon = .001$, $n = 12$.

Training Sequence:

$\mathbf{x}_1$ = (-.37449, .98719)	$\mathbf{x}_7$ = (-.59161, .17968)
$\mathbf{x}_2$ = (.63919,-.11875)	$\mathbf{x}_8$ = (.14093,1.76413)
$\mathbf{x}_3$ = (-.83293, .60645)	$\mathbf{x}_9$ = (.70898,-.35017)
$\mathbf{x}_4$ = (-.70534,-1.21856)	$\mathbf{x}_{10}$ = (.30038, .79836)
$\mathbf{x}_5$ = (-.28952,-.94821)	$\mathbf{x}_{11}$ = (.30165,1.06552)
$\mathbf{x}_6$ = (1.09924, .516)	$\mathbf{x}_{12}$ = (.37801,-.32708)

$\hat{A}_0 = \{(2,2), (2,-2), (-2,2), (-2,-2)\}$
$= \{\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \mathbf{y}_4\}$

D_{-1} = 9.99E + 62 (∞ on a microcomputer)

Set $m = 0$.

m=0 (1) Find $P(\hat{A}_0) = \{S_1, S_2, S_3, S_4\}$:

$\mathbf{x}_j \in S_i$ if $d(\mathbf{x}_j, \mathbf{y}_i) \leq d(\mathbf{x}_j, \mathbf{y}_m)$, all m.

$S_1 = \{\mathbf{x}_6, \mathbf{x}_8, \mathbf{x}_{10}, \mathbf{x}_{11}\}$

$S_2 = \{\mathbf{x}_2, \mathbf{x}_9\}$

$S_3 = \{\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_7\}$

$S_4 = \{\mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_{12}\}$

Compute D_0:

$$D_0 = \frac{1}{12} \sum_{j=1}^{12} \min_{\mathbf{y} \in \hat{A}_0} d(\mathbf{x}_j, \mathbf{y}) = 2.0172 .$$

(2) $(D_{-1} - D_0)/D_0 > .001$, continue.

(3) Find the optimal reproduction alphabet $\hat{A}_1 \triangleq \hat{\mathbf{x}}(P(\hat{A}_0)) = \{\hat{\mathbf{x}}(S_i), i=1, \ldots, 4\}$:

$\hat{\mathbf{x}}(S_1) = (\mathbf{x}_6 + \mathbf{x}_8 + \mathbf{x}_{10} + \mathbf{x}_{11})/4 = (.46055, 1.036)$

$\hat{\mathbf{x}}(S_2) = (\mathbf{x}_2 + \mathbf{x}_9)/2 = (.674085, -.23446)$

$\hat{\mathbf{x}}(S_3) = (\mathbf{x}_1 + \mathbf{x}_3 + \mathbf{x}_7)/3 = (-.599676, .591106)$

$\hat{\mathbf{x}}(S_4) = (\mathbf{x}_4 + \mathbf{x}_5 + \mathbf{x}_{12})/3 = (-.457623, -.831283)$

Set $m = 1$. Go to (1).

m=1 (1) Find $P(\hat{A}_1)$:

Evaluating distortions shows $P(\hat{A}_1) = P(\hat{A}_0)$ (no change in partition)

Compute D_1:

$$D_1 = \frac{1}{12} \sum_{j=1}^{12} \min_{\mathbf{y} \in \hat{A}_1} d(\mathbf{x}_j, \mathbf{y}) = .0987308 .$$

(2) $(D_0 - D_1)/D_1 \simeq 19 > .001$

(3) $\hat{A}_2 \triangleq \hat{\mathbf{x}}(P(\hat{A}_1)) = \hat{A}_1$, since $P(\hat{A}_1) = P(\hat{A}_0)$ and hence $\hat{\mathbf{x}}(P(\hat{A}_1)) = \hat{\mathbf{x}}(P(\hat{A}_0)) = \hat{A}_1$. Thus $\hat{A}_1$ is a fixed point. Set $m=2$. Go to (1).

m=2 (1) $P(\hat{A}_1) = P(\hat{A}_0)$ and hence $D_2 = D_1$ and hence $(D_1 - D_2)/D_2 = 0 < .001$. Halt with final quantizer described by $\{\hat{A}_1, P(\hat{A}_1)\}$.

Note: Characters with tildes underneath appear boldface in text.

In other words, if the quantizer stays the same for two iterations, then the two distortions are equal and an "$\epsilon = 0$" threshold is satisfied.

As a more realistic example, the algorithm was run for the scalar ($k = 1$) case with $N = 2, 3, 4, 6$ and 8, using a training sequence of 10,000 samples per quantizer output from a zero-mean, unit-variance memoryless Gaussian source. The resulting quantizer outputs and distortion were within 1% of the optimal values reported by Max [15]. No more than 20

Exhibit W Page 440

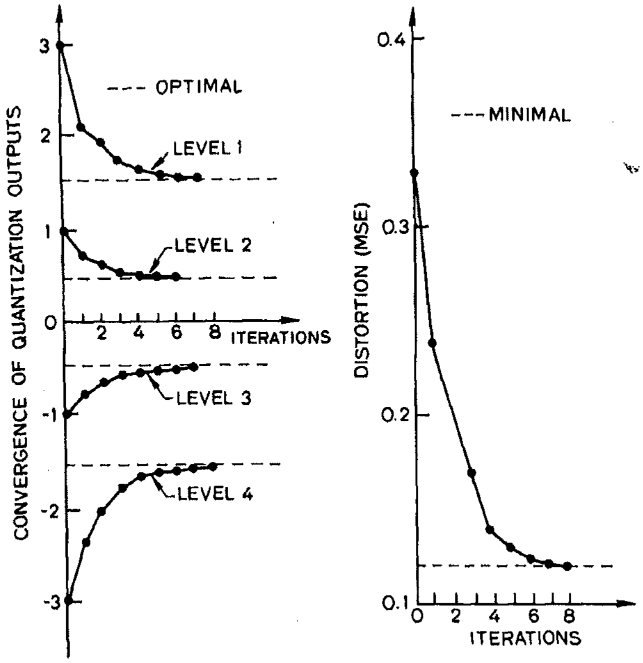


Fig. 1. The Basic Algorithm: Gaussian Source $N = 4$.

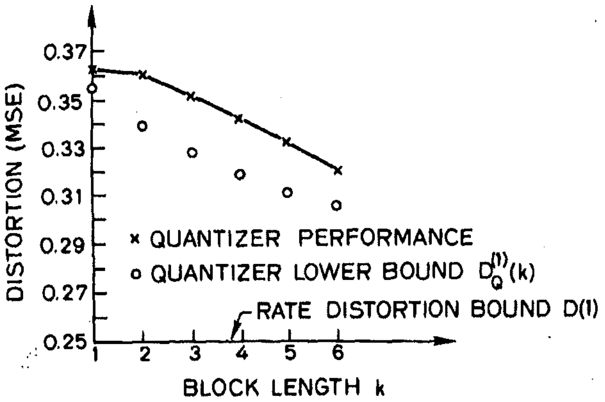

DISTORTION (MSE)
× QUANTIZER PERFORMANCE
o QUANTIZER LOWER BOUND $D_Q^{(1)}(k)$
RATE DISTORTION BOUND D(1)
BLOCK LENGTH k

Fig. 2. Block Quantization Rate 1 bit/symbol Gaussian Source.

iterations were required for $N = 8$ and, for smaller N, the number of iterations was considerably smaller. Figure 1 describes the convergence rate of one of the tests for the case $N = 4$.

The algorithm was then tried for block quantizers for memoryless Gaussian variables with block lengths k equal to 1, 2, 3, 4, 5 and 6 and a rate of one bit per sample, so that $N = 2^k$. The distortion criterion was again the squared-error distortion measure of (1). The algorithm used a training sequence of 100,000 samples. In each case, the algorithm converged in fewer than 50 iterations and the resulting distortion is plotted in Fig. 2, together with the one bit-per-symbol scalar case as a function of block length. For comparison, the rate-distortion bound [28, p. 99] $D(R) = 2^{-2R}$ for $R = 1$ bit-per-symbol is also plotted. As expected and as shown in Fig. 2, the block quantizers outperform the scalar quantizer, but for these block lengths, the performance is still far from the rate distortion bound (which is achievable, in principle, only in the limit as $k \to \infty$). A more favorable comparison is obtained using a recent result of Yamada, Tazaki, and Gray [29] which provides a lower bound to the performance of an optimal N-level k-dimensional quantizer with a difference distortion measure when N is large. This bound provides strict improvement over the rate-distortion bound for fixed k and tends to the rate-distortion bound as $k \to \infty$. In the current case, the bound has the form

$$D_Q^{(k)}(R) = D(R) \cdot \left\{ \left(\frac{e}{1 + k/2} \right) \Gamma(1 + k/2)^{2/k} \right\},$$

where Γ is the gamma function. This bound is theoretically inappropriate for small k, yet it is surprisingly close for the $k = 1$ result, which is known to be almost optimal. For $k = 6$, $N = 2^6 = 64$ is moderately large and the closeness of the actual performance to the lower bound, compared to optimal performance provided by $D_Q^{(k)}(1)$, suggests that the algorithm is indeed providing a quantizer with block length six and rate one bit-per-symbol that is nearly optimal (within 6% of the optimal).

LLOYD'S EXAMPLE

Lloyd [1] provides an example where both variational and fixed-point approaches can yield locally optimal quantizers instead of a globally optimum quantizer. We next propose a slight modification of the fixed-point algorithm that indeed finds a globally optimum quantizer in Lloyd's example. We conjecture that this technique will work more generally, but we have been unable to prove this theoretically. A similar technique can be used with the stationary-point algorithm.

Instead of using samples from the source that we wish to quantize, we use samples corrupted by additive independent noise, where the marginal distribution of the noise is such that only one locally optimum quantizer exists for it. As an example, for scalar quantization with the squared-error distortion measure, we use Gaussian noise. In this case, any locally optimal quantizer is also globally optimum. Other distributions, such as the uniform or a discrete amplitude noise with an alphabet size equal to the number of quantizer output levels, can also be used.

When the noise power is much greater than the source power, the distribution of their sum is essentially the distribution of the noise. We assume that, initially, the noise power is so large that only one locally optimum quantizer exists for the sum; hence, regardless of the initial guess, the algorithm will converge to this optimum. On the next step, the noise power is reduced slightly and the quantizer resulting from the previous run is used as the initial guess. Intuitively, since the noise has been reduced by a small amount, the global optimum for the new sum should be close to that of the previous sum (we use the same source and noise samples with reduced noise power). Thus we expect that the algorithm will converge to the global optimum even though new local optimum points might have been introduced. We continue in the same manner reducing the noise gradually to zero.

Exhibit W Page 441

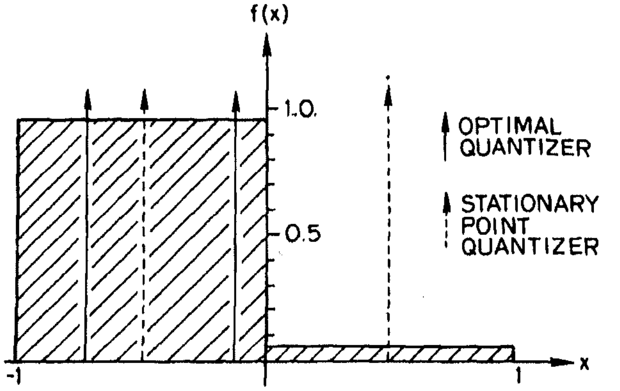


Fig. 3. The Probability Density Function.

To illustrate how the algorithm works, we use a source with a probability density function as shown in Fig. 3. In this case, there are two locally optimum two-level quantizers. One has the output levels +0.5 and −0.5 and yields a mean-squared error of 0.083, the second (which is the global optimum) has output levels −0.71 and 0.13 and yields a mean-squared error 0.048. (This example is essentially the same as one of Lloyd's [1].)

The modified algorithm was tested on a sequence of 2,000 samples chosen according to the probability density shown in Fig. 3. Gaussian noise was added starting at unity variance and reducing the variance by approximately 50% on each successive run. The initial guess was + 0.5 and − 0.5 which is the non-global optimum. Each run was stopped when the distortion was changed by less than 0.1% from its previous value.

The results are given in Fig. 4 and it is seen that, in spite of the bad initial guess, the modified algorithm converges to the globally optimum quantizer.

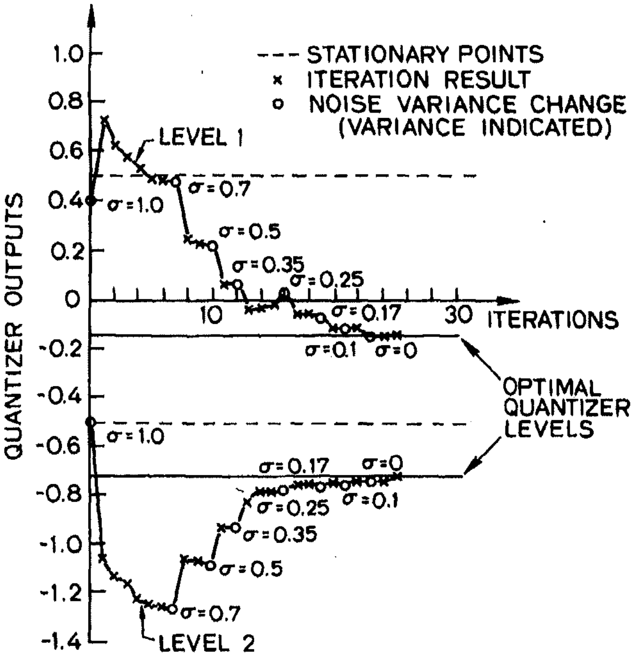


Fig. 4. The Modified Algorithm.

SPEECH EXAMPLE

In the next example, we consider the case of a speech compression system consisting of an LPC analysis of 20 ms-long speech frames producing a voiced/unvoiced decision and a pitch, a gain, and a normalized inverse filter as previously described, followed by quantization where the pitch and gain are separately quantized as usual, but the normalized filter coefficients $(a_1, \cdots, a_K) = (x_0, x_1, \cdots, x_{K-1})$ are quantized as a vector with $K = 10$, using the distortion measure of (7)-(8). The training sequence consisted of a sequence of normalized inverse filter parameter vectors[3]. The LPC analysis was digital, and hence the training sequence used was already "finely quantized" to 10 bits per sample or 100 bits for each vector. The original gain required 12 bits per speech frame and the pitch used 8 bits per speech frame. The total rate of the LPC output (which we wish to further compress by block quantization) is 6000 bits/s. No further compression of gain or pitch was attempted in these experiments as our goal was to study only potential improvement when block quantizing the normalized filter parameters. The more complete problem including gain and pitch involves many other issues and is the subject of a paper still in preparation.

For the distortion measure of (7)-(8), the centroid of a subset S of a training sequence $\{x_j, j = 0, 1, \cdots, n-1\}$ is the vector u minimizing

$$\sum_{j:x_j \in S} (x_j - u) R(x_j) (x_j - u)^t.$$

We observe that the autocorrelation matrix $R(x)$ is a natural byproduct of the LPC analysis and need not be recomputed. This minimization, however, is a minimum-energy-residual minimization problem in LPC analysis and it can be solved by standard LPC algorithms such as Levinson's algorithm [10]. Alternatively, it is a much studied minimization problem in Toeplitz matrix theory [30] and the centroid can be shown via variational techniques to be

$$\hat{x}(S) = \left\{ \sum_{j:x_j \in S} R(x_j) \right\}^{-1} \sum_{j:x_j \in S} R(x_j) x_j^t. \tag{15}$$

The splitting technique for the initial guess and a distortion threshold of 0.5% were used. The complete algorithm for this example can thus be described as follows:

(0) Initialization: Fix $N = 2^R$, R an integer, where N is the largest number of levels desired. Fix $K = 10$, n = length of training sequence, $\epsilon = .005$. Set $M = 1$.

Given a training sequence $\{x_j; j = 0, \cdots, n-1\}$, set $A = \{x_j; j = 0, \cdots, n-1\}$, the training sequence alphabet. Define $\hat{A}(1) = \hat{x}(A)$, the centroid of the entire training sequence using (15) or Levinson's algorithm.

(3) The training sequence and additional test data of LPC reflection coefficients were provided by Signal Technology Inc. of Santa Barbara and were produced using standard LPC techniques on a single male speaker.

Exhibit W Page 442

(1) (Splitting): Given $\hat{A}(M) = \{y_i, i = 1, \cdots, M\}$, split each reproduction vector y_i into $y_i + \epsilon$ and $y_i - \epsilon$, where ϵ is a fixed perturbation vector. Set $\hat{A}_0(2M) = \{y_i + \epsilon, y_i - \epsilon, i = 1, \cdots, M\}$ and then replace M by $2M$.

(2) Set $m = 0$ and $D_{-1} = \infty$.

(3) Given $\hat{A}_m(M) = \{y_1, \cdots, y_M\}$, find its optimum partition $P(\hat{A}_m(M)) = \{S_i; i = 1, \cdots, M\}$, that is, $x_j \in S_i$ if $d(x_j, y_i) \leq d(x_j, y_l)$, all l. Compute the resulting distortion

$$D_m = D(\{\hat{A}_m(M), P(\hat{A}_m(M))\})$$

$$= n^{-1} \sum_{j=0}^{n-1} \min_{y \in \hat{A}_m} d(x_j, y).$$

(4) If $(D_{m-1} - D_m)/D_m \leq \epsilon = .005$, then go to step (6). Otherwise continue.

(5) Find the optimal reproduction alphabet $\hat{A}_{m+1}(M) = \hat{x}(P(\hat{A}_m(M)) = \{\hat{x}(S_i); i = 1, \cdots, 1\}$ for $P(\hat{A}_m(M))$. Replace m by $m + 1$ and go to (3).

(6) Set $\hat{A}(M) = \hat{A}_m(M)$. The final M-level quantizer is described by $\hat{A}(M)$. If $M = N$, halt with final quantizer described by $\hat{A}(N)$. Otherwise go to step (1).

Table 2 describes the results of the algorithm for $N = 64$, and hence for one- to eight-bit quantizers trained on $n =$ 19,000 frames of LPC speech produced by a single speaker. The distortion at the end of each iteration is given and, in all cases, the algorithm converged in fewer than 14 iterations. When the resulting quantizers were applied to data from the same speaker outside of the training sequence, the resulting distortion was within 1% of that within the training sequence. A total of three and one-half hours of computer time on a PDP 11/35 was required to obtain all of these codebooks.

Figure 5 depicts the rate of convergence of the algorithm with a training sequence length for a 16-level quantizer. Note the marked difference between the distortion for 2400 frames inside the training sequence and outside the training sequence for short training sequences. For a long training sequence of over 12,000 frames, however, the distortion is nearly the same.

Tapes of the synthesized speech at 8 bits per frame for the normalized model sounded similar to those of the original LPC speech with 100 bits per frame for the normalized model (the gain and the pitch were both left at the original LPC rate of 12 and 8 bits per frame, respectively). While extensive subjective tests were not attempted, all informal listening tests judged the synthesized speech perfectly intelligible (when heard *before* the original LPC!) and the quality only slightly inferior when the two were compared. The overall compression was from 6000 bits/s to 1400 bits/s. This is not startling as existing scalar quantizers that optimally allocate bits among the parameters and optimally quantize each parameter using a spectral deviation distortion measures [11] also perform well in this range. It is, however, promising as these were preliminary results with no attempt to further compress pitch and gain (which, taken together in our system, had more than twice the bit rate of the normalized model vector quantizer). Further results on applications of the algorithm to the overall speech compression system will be the subject of a forthcoming paper [33].

TABLE 2
ITAKURA-SAITO DISTORTION VS. NUMBER OF ITERATIONS. TRAINING SEQUENCE LENGTH = 19,000 FRAMES.

NUMBER OF LEVELS	DISTORTION	ITERATION NUMBER
2	10.33476	1
	1.98925	2
	1.78301	3
	1.67244	4
	1.55983	5
	1.49814	6
	1.48493	7
	1.48249	8
4	1.38765	1
	1.07906	2
	1.04223	3
	1.03252	4
	1.02709	5
8	0.96210	1
	0.85183	2
	0.81353	3
	0.79191	4
	0.77472	5
	0.76188	6
	0.75130	7
	0.74383	8
	0.73341	9
	0.71999	10
	0.71346	11
	0.70908	12
	0.70578	13
	0.70347	14
16	0.64653	1
	0.55665	2
	0.51810	3
	0.50146	4
	0.49235	5
	0.48761	6
	0.48507	7
32	0.44277	1
	0.40452	2
	0.39388	3
	0.38667	4
	0.38128	5
	0.37778	6
	0.37574	7
	0.37448	8
64	0.34579	1
	0.31766	2
	0.30850	3
	0.30366	4
	0.30086	5
	0.29891	6
	0.29746	7
128	0.27587	1
	0.25628	2
	0.24928	3
	0.24550	4
	0.24309	5
	0.24142	6
	0.24021	7
	0.23933	8
256	0.22458	1
	0.20830	2
	0.20228	3
	0.19849	4
	0.19623	5
	0.19479	6
	0.19386	7
	0.19319	8

Exhibit W Page 443

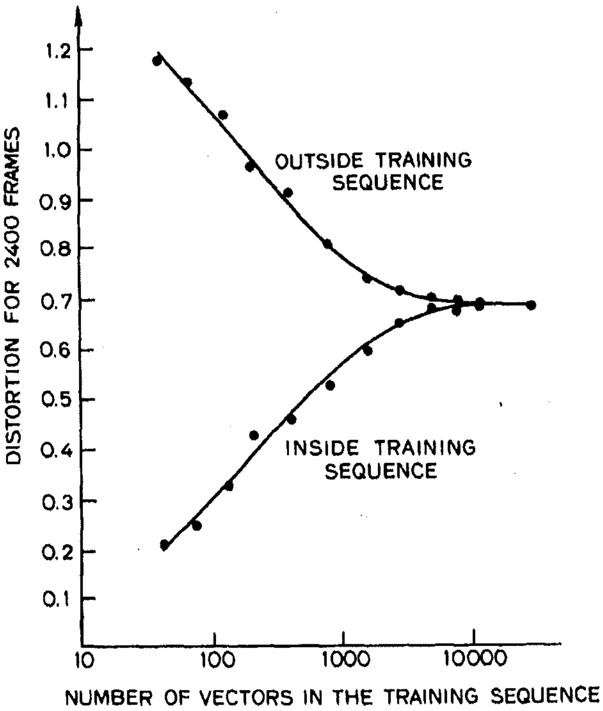

DISTORTION FOR 2400 FRAMES
1.2
1.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
OUTSIDE TRAINING SEQUENCE
INSIDE TRAINING SEQUENCE
10
100
1000
10000
NUMBER OF VECTORS IN THE TRAINING SEQUENCE

Fig. 5. Convergence with Training Sequence.

EPILOGUE

The Gaussian example of Figure 2, Lloyd's example, and the speech example were run on a PDP 11/34 minicomputer at the Stanford University Information Systems Laboratory. The simple example of Table 1 was run in BASIC on a Cromemco System 3 microcomputer. As a check, the microcomputer program was also used to design quantizers for the Gaussian case of Figure 2 using the splitting method, $k = 1$, 2, and 3, and a training sequence of 10,000 vectors. The results agreed with the PDP 11/34 run to within one percent.

ACKNOWLEDGMENT

The authors would like to acknowledge the help of J. Markel of Signal Technology, Inc. of Santa Barbara and A. H. Gray, Jr., of the University of California at Santa Barbara in both the analysis and synthesis of the speech example.

REFERENCES

[1] Lloyd, S. P., "Least Squares Quantization in PCM's," Bell Telephone Laboratories Paper, Murray Hill, NJ, 1957.

[2] Gray, R. M., J. C. Kieffer and Y. Linde, "Locally Optimal Block Quantization for Sources without a Statistical Model," Stanford University Information Systems Lab Technical Report No. L-904-1, Stanford, CA, May 1979 (submitted for publication).

[3] Itakura, F. and S. Saito, "Analysis Synthesis Telephony Based Upon Maximum Likelihood Method," *Repts. of the 6th Internat'l. Cong. Acoust.*, Y. Kohasi, ed., Tokyo, C-5-5, C17-20, 1968.

[4] Itakura, F., "Maximum Prediction Residual Principle Applied to Speech Recognition," *IEEE Trans. ASSP*, 23, pp. 67-72, Feb. 1975.

[5] Chaffee, D. L., "Applications of Rate Distortion Theory to the Bandwidth Compression of Speech Signals," Ph.D. Dissertation, Univ. of Calif. at Los Angeles, 1975.

[6] Gray, R. M., A. Buzo, A. H. Gray, Jr., and J. D. Markel, "Source Coding and Speech Compression," *Proc. of the 1978 Internat'l. Telemetering Conf.*, pp. 371-878, 1978.

[7] Matsuyama, Y., A. Buzo and R. M. Gray, "Spectral Distortion Measures for Speech Compression," Stanford Univ. Inform. Systems Lab. Tech. Rept. 6504-3, Stanford, CA, April, 1978.

[8] Buzo, A., "Optimal Vector Quantization for Linear Predicted Coded Speech," Stanford Univ., Ph.D. Dissertation, Dept. of Elec. Engrg., August, 1978.

[9] Matsuyama, Y. A., "Process Distortion Measures and Signal Processing," Ph.D. Dissertation, Dept. of Elec. Engrg., Stanford Univ., 1978.

[10] Markel, J. D. and A. H. Gray, Jr., *Linear Prediction of Speech*, Springer-Verlag, NY 1976.

[11] Gray, A. H., Jr., R. M. Gray and J. D. Markel, "Comparison of Optimal Quantizations of Speech Reflection Coefficients," *IEEE Trans. ASSP*, Vol. 24, pp. 4-23, Feb. 1977.

[12] Dalenius, T., "The Problem of Optimum Stratification," *Skandinavisk Aktuuvielidskrift*, Vol. 33, pp. 203-213, 1950.

[13] Fisher, W. D., "On a Pooling Problem from the Statistical Decision Viewpoint," *Econometrica*, Vol. 21, pp. 567-585, 1953.

[14] Cox, D. R., "Note on Grouping," *J. of the Amer. Statis. Assoc.*, Vol. 52, pp. 543-547, 1957.

[15] Max, J., "Quantizing for Minimum Distortion," *IRE Trans. on Inform. Theory*, IT-6, pp. 7-12, March 1960.

[16] Fleischer, P., "Sufficient Conditions for Achieving Minimum Distortion in a Quantizer," *IEEE Int. Conv. Rec.*, pp. 104-111, 1964.

[17] Luenberger, D. G., *Optimization by Vector Space Methods*, John Wiley & Sons, NY, 1969.

[18] Luenberger, D. G., *Introduction to Linear and Nonlinear Programming*, Addison-Wesley, Reading, MA, 1973.

[19] Rockafellar, R. T., *Convex Analysis*, Princeton Univ. Press, Princeton, NJ, 1970.

[20] Zador, P., "Topics in the Asymptotic Quantization of Continuous Random Variables," Bell Telephone Laboratories Technical Memorandum, Feb. 1966.

[21] Gersho, A., "Asymptotically Optimal Block Quantization," *IEEE Trans. on Inform. Theory*, Vol. IT-25, pp. 373-380, 1979.

[22] Chen, D. T. S., "On Two or More Dimensional Optimum Quantizers," *Proc. 1977 IEEE Internat'l. Conf. on Acoustics, Speech, & Signal Processing*, pp. 640-643, 1977.

[23] Caprio, J. R., N. Westin and J. Esposito, "Optimum Quantization for Minimum Distortion," *Proc. of the Internat'l. Telemetering Conf.*, pp. 315-323, Nov. 1978.

Exhibit W Page 444

[24] Menez, J., F. Boeri, and D. J. Esteban, "Optimum Quantizer Algorithm for Real-Time Block Quantizing," *Proc. of the 1979 IEEE Internat'l. Conf. on Acoustics, Speech, & Signal Processing*, pp. 980-984, 1979.

[25] MacQueen, J., "Some Methods for Classification and Analysis of Multivariate Observations," *Proc. of the Fifth Berkeley Symposium on Math., Stat. and Prob.*, Vol. 1, pp. 281-296, 1967.

[26] Ball, G. H. and D. J. Hall, "Isodata—An Iterative Method of Multivariate Analysis and Pattern Classification," in *Proc. IFIPS Congr.*, 1965.

[27] Levinson, S. E., L. R. Rabiner, A. E. Rosenberg and J. G. Wilson, "Interactive Clustering Techniques for Selecting Speaker-Independent Techniques for Selecting Speaker-Independent Reference Templates for Isolated Word Recognition," *IEEE Trans. ASSP*, Vol. 27, pp. 134-141, 1979.

[28] Berger, T., *Rate Distortion Theory*, Prentice-Hall, Englewood Cliffs, NJ, 1971.

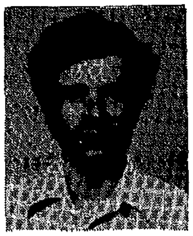
[29] Yamada, Y., S. Tazaki and R. M. Gray, "Asymptotic Performance of Block Quantizers with Difference Distortion Measures," to appear, *IEEE Trans. on Inform. Theory*.

[30] Grenander, U. and G. Szego, *Toeplitz Forms and Their Applications*, Univ. of Calif. Press, Berkeley, 1958.

[31] Forgy, E., "Cluster Analysis of Multivarate Data: Efficiency vs. Interpretability of Classifications," Abstract, *Biometrics*, Vol. 21, p. 768, 1965.

[32] Chaffee, D. L. and J. K. Omura, "A Very Low Rate Voice Compression System," Abstract in *Abstracts of Papers, 1974, IEEE Intern. Symp. on Inform. Theory*, Notre Dame, Oct. 28-31, IEEE, 1974.

[33] Buzo, A., A. H. Gray, Jr., R. M. Gray, and J. D. Markel, "Speech Coding Based on Vector Quantization," submitted for publication.

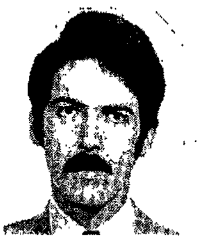


Yoseph Linde (S'76–M'78) was born in Sendzislaw, Poland on June 14, 1947. He received the B.Sc. degree from the Technion, Israel Institute of Technology in 1970, the M.Sc. degree from the Tel-Aviv University in 1975 and the Ph.D. degree from Stanford University, Stanford, California, in 1977, all in Electrical Engineering.

From 1970 to 1975 he was with the Signal Corps., Israeli Defense Forces, where he was involved in research and development of military communications systems. In 1976 and 1977 he was a research assistant at the Information Systems Laboratory at Stanford University involved in research in data compression systems, in particular tree and trellis codes.

Dr. Linde is currently with Codex Corporation, Mansfield, Massachusetts where he is involved in research and development in the areas of digital signal processing, modems and data networks.

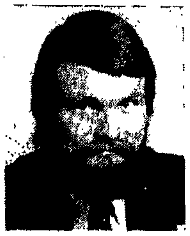
★



Andrés Buzo (S'76–M'78) was born in Mexico City, Mexico, on November 30, 1949. He received the electrical and mechanical engineer degree from the National University of Mexico, Mexico City, in 1974, and the M.S. and Ph.D. degrees from Stanford University, Stanford, California in 1975 and 1978, respectively.

In 1978 he was at Signal Technology in Santa Barbara, California. Now he is at the Instituto de Ingenieria of the National University of Mexico where he is engaged in research on digital signal processing of speech signals and data compression.

★



Robert M. Gray (S'68–M'69–SM'77) was born in San Diego, California, on November 1, 1943. He received the B.S. and M.S. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, in 1966, and the Ph.D. degree from the University of Southern California, Los Angeles, in 1969.

Since 1969 he has been with the Electrical Engineering Department and the Information Systems Laboratories, Stanford University, Stanford, California, where he is engaged in teaching and research in communications and information theory.

Dr. Gray is a member of Sigma Xi, Eta Kappa Nu, the Mathematical Association of America, the Society for Industrial and Applied Mathematics, the Institute of Mathematical Statistics, the American Mathematical Society, and the Société des Ingénieurs et Scientifiques de France. He has been a member of the Board of Governors of the IEEE Professional Group on Information Theory since 1975 and an Associate Editor of the IEEE TRANSACTIONS ON INFORMATION THEORY since September 1977.

Exhibit W Page 445